МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ОБ АУДИТЕ**

Выполнил студент группы 22/2                                       С.А. Махалин

Направление подготовки  01.03.02  Прикладная математика и информатика Курс    2

Отчет приняла кандидат физико-математических наук, доцент                                                                                       Е.П. Лукащик

Краснодар

2022 г.

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc98113292)

[2 Защита от XSS 3](#_Toc98113293)

[3 Защита от SQL-INJECTION 3](#_Toc98113294)

[4 Защита от CSRF 4](#_Toc98113295)

[5 Защита от Upload и Include уязвимости 4](#_Toc98113296)

[6 Вывод 4](#_Toc98113297)

# 1 Цель работы

Провести аудит безопасности вашего приложения и исправить уязвимости. В нем должны быть разделы, посвященные уязвимостям XSS, SQL Injection, CSRF, Include, Upload. В отчете указать по каждой уязвимости примененные методы защиты с примерами вашего кода.

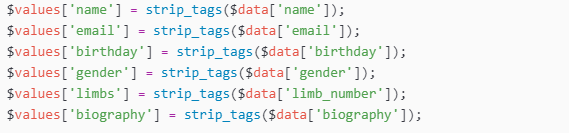
# 2 Защита от XSS

**XSS**– это сокращение понятия расшифровываемое как "межсайтовый скриптинг". В задачи межсайтового скриптинга и главной целью XSS является получение **cookies** пользователей атакуемого сайта путем встраивания вредоносного кода в тело HTML страницы. Является разновидностью атаки «Внедрение кода».

Изначально, при загрузке в форму данные из базы не приводились к безопасному формату и была возможность провести XSS-атаку:

****

Используем strip\_tags() (удаляет HTML и PHP тэги) для строковых данных и intval() (Возвращает целое значение переменной var) для целочисленных:

****

# 3 Защита от SQL-INJECTION

SQL Injection - метод, предназначенный для введения SQL запросов/команд через web-страницы. Многие web-страницы используют параметры, представленные Web пользователям, и делают SQL запрос базы данных. С использованием SQL Injection можно послать придуманное имя пользователя и/или поле пароля, изменяющее SQL запрос.

Необходимо экранировать символы поступающие из формы призаписав базу данных (DBO::quote) или использовать подготовленные запросы:

# 

# 4 Защита от CSRF

[CSRF](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%B4%D0%B5%D0%BB%D0%BA%D0%B0_%D0%BC%D0%B5%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D1%8B%D1%85_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2) (Cross-Site Request Forgery, также XSRF) – опаснейшая атака, которая приводит к тому, что хакер может выполнить на неподготовленном сайте массу различных действий от имени других, зарегистрированных посетителей. Использует недостатки протокола HTTP.

Защитим важные поля форм, (изменение записи в базе данных и удаление записи) добавив токен, привязанный к сессии пользователя и проверку токена:

# 

# 5 Защита от Upload и Include уязвимости

PHP-include — уязвимость, которая позволяет «принудить» произвольный файл и выполнить PHP код в любом файле на сервере.

Upload уязвимостей нет, т. к. пользователь не загружает на сервер файлы. Include уязвимостей нет, т. к. пользователь не дает данные для подключения модулей обработчиков.

# 6 Вывод

Таким образом мы защитили сайт от различного рода уязвимостей в коде, атак типа внедрения SQL и атак на посетителей веб-сайта. Теперь при загрузке в форму данные из базы приводятся к безопасному формату.